ECS132

Weili Yin

HW2

1: 1: $We\ know\ Var(X+Y) = Var(X) + Var(Y) + 2Cov(X,Y); Replace\ Y\ with\ -Y, we\ have\ Var(X) + Var(-Y) + 2Cov(X,-Y); \because Var(-Y) = Var(Y), Cov(X,-Y) = E[-XY] - E[X]E[-Y] = -E[XY] + E[X]E[Y] = -Cov(X,Y);$

$Therefore, Var(X-Y) = Var(X) + Var(Y) - 2Cov(X,Y)$

$and\ X,Y\ are\ independent, Cov(X,Y) = 0. Thus, Var(X-Y) = Var(X) + Var(Y)$

2: $Var(L_1) = E[L_1^2] - E^2(L_1) = (\sum L_1^2 * P) - E^2(L_1) = .8 - .36 = .44$

$Var(L_2) = E[L_2^2] - E^2(L_2);$

$P(L_2 = 0) = .5^2 + .4 * .2 * .5 + .1 * .2^2 * .5 = .292$

$P(L_2 = 1) = .5 * .4 + .4 * .8 * .5 + .4 * .2 * .4 + 2(.1 * .2 * .8 * .5) + .1 * .2^2 * .4 = .4096$

$P(L_2 = 2) = .5 * .1 + .4 * .8 * .4 + .4 * .2 * .1 + .1 * .8^2 * .5 + 2(.1 * .2 * .8 * .4) + .1 * .2^2 * .1 = .2312$

$P(L_2 = 3) = .4 * .8 * .1 + .1 * .8^2 * .4 + 2(.1 * .8 * .2 * .1) = .0608$

$P(L_2 = 4) = .1 * .8^2 * .1 = .0064$

$Var(L_2) = E[L_2^2] - E^2(L_2) = (1 * .4096 + 4 * .2312 + 9 * .0608 + 16 * .0064) - (1 * .4096 + 2 * .2312 + 3 * .0608 + 4 * .0064)^2 = .817$

$P(L_2 - L_1 = -2) = .1 * .2^2 * .5 = .002$

$P(L_2 - L_1 = -1) = .4 * .2 * .5 + .1 * .2 * .8 * .5 * 2 + .1 * .2^2 * .4 = .0576$

$P(L_2 - L_1 = 0) = .5^2 + .4 * .8 * .5 + .4 * .2 * .4 + .1 * .8^2 * .5 + .1 * .2 * .8 * .4 * 2 + .1 * .2^2 * .1 = .4872$

$P(L_2 - L_1 = 1) = .5 * .4 + .4 * .8 * .1 + .1 * .8^2 * .1 = .0884$

$Var(L_2 - L_1) = E[(L_2 - L_1)^2] - E^2[L_2 - L_1] = .05536$

Simulation: Var(L1):

```
> ls1 <- sample(0:2,10000,prob = c(.5,.4,.1),replace = TRUE)
> var(ls1)
[1] 0.4410401
```

Var(L2):

```
getL2 <- function(){
  L1 <- sample(0:2,1,prob = c(.5,.4,.1))
  NumofPeople <- L1
  i <- 0
  while (i<NumofPeople) {
    if(runif(1) < .2){ #quit one people
      L1 <- L1-1
    }
    i <- i+1
  }
  return(L1+sample(0:2,1,prob=c(.5,.4,.1)))
}
getL2list <- function(num){
  list <- c()
  i <- 0
  while (i < num) {
    list <- c(list ,getL2())
    i <- i+1
  }
  return(list)
}
```

```
> var(getL2list(10000))
[1] 0.82005
```

Var(L2-L1)

```
getL2L1 <-function(){
  L1 <- sample(0:2,1,prob = c(.5,.4,.1))
  NumofPeople <- L1
  initL1 <- L1
  i <- 0
  while (i<NumofPeople) {
    if(runif(1) < .2){ #quit one people
      L1 <- L1-1
    }
    i <- i+1
  }
  return(L1+sample(0:2,1,prob=c(.5,.4,.1))-initL1)
}
getL2L1list <- function(num){
  list <- c()
  i <- 0
  while (i < num) {
    list <- c(list ,getL2L1())
    i <- i+1
  }
  return(list)
}
```

```
> var(getL2L1list(10000))
[1] 0.5539962
```

2: $P(3) = \text{HHH} = .5^3; P(4) = THHH = .5^4; P(5) = anyTHHH = .5^4; P(6) = 1 - P(3) - P(4) - P(5) = .75$

$E(X) = 3 * P(3) + 4 * P(4) + 5 * P(5) + 6 * P(6) = 5.4$

Simulation:

```
toss <- function(){
  headaccu <- 0
  round <- 0
  testls <- c()
  while (headaccu < 3 & round < 6) {
    if(runif(1) <.5){ #get HEAD
      headaccu <- headaccu+1
      testls <- c('H',testls)
    }else{
      headaccu <- 0
      testls <- c('T',testls)
    }
    round <- round +1
  }
  #print(testls)
  return(round)
}
gettosslist <- function(num){
  list <- c()
  i <- 0
  while (i < num) {
    list <- c(list ,toss())
    i <- i+1
  }
  return(list)
}
```

```
> mean(gettosslist(10000))
[1] 5.4575
```

3:

$W = \sum_1^7 |X_{i+1} - X_i|$

```
library(gtools)
fir <- function(z){z[1]}
permn <- function(x,m,FUN){
  ls <- permutations(n = length(x), r = m, x)
  rs <- c()
  count <- length(ls)/m
  for (i in 1:count) {
    rs <- c(rs,FUN(ls[i,]))
  }
  return(rs)
}

### Below are for P3.2
fun <- function(list){
  sum <- numeric(1)
  sum <- 0
  i <- 1
  while (i < 8) {
    sum <- sum + abs(list[i+1]-list[i])
    i <- i+1
  }
  return(sum)
}
```

To get EW, we have function: that take all possible permutations and calculate sum, Then use mean function to get the expected value, the result is around 30.333

```
> sum(permn(1:12,8,fun))/length(permn(1:12,8,fun))
[1] 30.33333
```

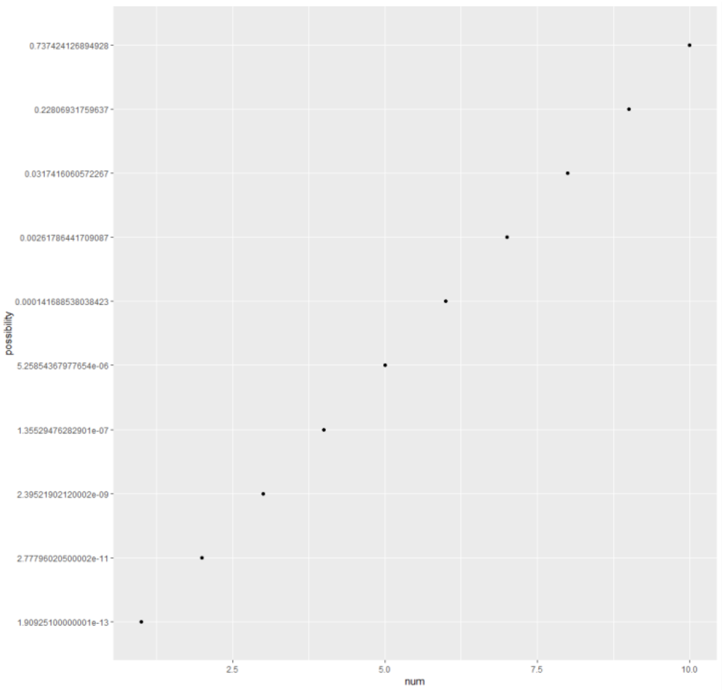
4: 1: $X \in \{2:8\}$

2: $$Pmf(X=x) = \begin{cases} \frac{1}{16}(x=2) \\ \frac{2}{16}(x=3) \\ \frac{3}{16}(x=4) \\ \frac{4}{16}(x=5) \\ \frac{3}{16}(x=6) \\ \frac{2}{16}(x=7) \\ \frac{1}{16}(x=8) \end{cases}$$

3: $$E(X) = \sum xpmf(x) = \frac{2}{16} + \frac{6}{16} + \frac{12}{16} + \frac{20}{16} + \frac{18}{16} + \frac{14}{16} + \frac{8}{16} = 5$$

5: 1: use ggplot2 to generate plot graph

```
###P5
library(ggplot2)
drawplot <-function(){
  num <- c(1:10)
  possibility <- NA
  sum <- 0
  for (i in 1:10) {
    possibility[i] <- dbinom(i,10,.97)
  }
  df <- data.frame(num,possibility)
  df$possibility <- as.factor(df$possibility)
  head(df)
  ggplot(df, aes(x=num, y=possibility)) + geom_point()
}
drawplot()
```



2: $P(fail \geq 2) = 1 - P(fail < 2)$

$P(fail < 2) = P(fail = 1) +$

$P(fail = 0) = .03 * .97^9 + .97^{10} = .76$